Guide to CodeWarrior Java for CS100

Computer Science, Cornell University

Spring 2000

1 Introduction

The main software for CS100 programming assignments is Metrowerks CodeWarrior.

- **Version**

  Version Pro 5 is installed in the CIT labs for Spring 2000.

  > *Earlier versions Pro 3 and Pro 4 can probably be used, but be aware that there are some differences in how they operate, e.g. some menus and dialog boxes are different, and older PC versions require MS Internet Explorer.*

- **Mac or PC**

  CodeWarrior Pro is available for both **Macintosh** and **Windows/NT**. It works the same in both environments, except for some minor differences due to the Windows and Macintosh user interfaces.

  Naturally, if you wish to use your floppy disks on both kinds of machines, you will need to choose a format that both can read: You should format your floppy disks as PC disks because the Mac can read the **PC format**, but not vice versa.

  We assume you already know how to use Macintosh or Windows to, for example, create and print a document using a word processor.

- **Public Labs**

  CIT has installed CodeWarrior in all of its labs that have machines that can support it. The only difference between the CIT version and the regular one is that some additional files have been added for simple Java programs, particularly for CS100.

- **CodeWarrior At Home: CUCS Stationery Extensions**

  If you have your own copy of CodeWarrior, you should add the `CUCSJavaApplication` and `CUCSJavaGraphicsApplication` stationery to it. The course home page should contain instructions for accessing and installing these files.

  Files are in standard formats for the web: self-extracting BinHex archives for Macintosh and zip files for Windows. Installation is simple. Unpack the files. On the Macintosh, drag the new additional stationery folders to folder `(Project Stationery):Java`; on Windows, the new stationery should be placed in CodeWarrior's `Stationery\Java` folder. Instructions are included with the files.

- **Formatting Conventions in this guide**

  In this handout, a menu command is given in the form "`name|command`", where `name` is the name of the menu and `command` is a command within it. For example, "select `Project|Run`" means select command `Run` from menu `Project`.

2 Basics of Files

- **Program/Source and Project Files**

  CodeWarrior uses two major kinds of files: *program* or *source files* contain Java program code, while *project files* contain additional information CodeWarrior needs to run the program. Each program has one project file and one or more source files.

  By convention, CodeWarrior names of project files end in "`.mcp`" (standing for Metrowerks CodeWarrior Project). Names of Java source files must end in "`.java`".

  > **IMPORTANT:** *CodeWarrior can get confused (e.g. be unable to find a file) if filenames contain spaces or any punctuation other than underlines, dashes, and periods.*

- **Backups**

  Remember to save your work frequently, not just when you are done for the day. You should also keep a second disk as a backup in case your first disk becomes corrupted or unreadable. You should copy your main floppy disk to the backup regularly.

  > **IMPORTANT:** *You **must** keep your files on your own disks. User files are removed from the public machines regularly.*

- **Saving Files before Running Programs or Quitting CodeWarrior**

  CodeWarrior does take a few simple precautions to protect your work. Before CodeWarrior runs your program, it saves all your files (unless you turned off this option). And when you are done working and leave the CodeWarrior environment by selecting `File`|`Quit`, if there are any open files that you haven't saved yet, you will be asked if you want to save them now. Click `Save` or `Don't Save`.

- **Reclaiming Disk Space**

  A lot of the space used on your disk isn't needed after you've finished a CodeWarrior project. From within CodeWarrior, with the project open, you may select `Project`|`Remove Object Code` to delete compiled Java machine code. At the desktop, inside the folder containing your project, you may delete file `JavaClasses.zip`, which also contains compiled Java machine code, as well as the folder that has the same name as the project file followed by `Data`, which contains information used by CodeWarrior to run the project. These items (object code, zip file, Data folder) can easily be recreated by running the program again.

3 (Finding and) Starting CodeWarrior

- **Macintosh.**

  In the CIT labs, CodeWarrior is buried inside several layers of folders. If the hard drive window is not open, double-click the hard drive icon. Then open the following succession of folders to find the CodeWarrior application: `Applications`; `CodeWarrior`; `Metrowerks CodeWarrior`. (Use `Option`-double-click to reduce clutter on the desktop by closing unneeded windows as you pass through them.) Start CodeWarrior by double-clicking the `CodeWarrior IDE` application icon.

- **Windows.**

  Click `Start`. From the resulting menu, select `Programs`|`CodeWarrior`|`CodeWarrior IDE` to start CodeWarrior. (On some machines, the `CodeWarrior IDE` may be in a menu named `Programs`|`Metrowerks CodeWarrior`.)

When CodeWarrior finishes launching, its menu bar will appear at the top of the screen. In the public labs, a dialog box might be displayed to remind you that this copy of CodeWarrior can be used only in the CIT labs. Click `OK` to continue.

4 Opening Project and Program Files

- **Opening an Existing Project**

  To use a project that you get from us, or to continue working on one you created earlier, double-click the project file icon. This will automatically start CodeWarrior if it is not already running. **Don't** double-click a `.java` file icon. This may open a different programming environment. Moreover, a project file must be open to run a program.

- **Opening Projects Inside CodeWarrior**

  To open a project from within CodeWarrior, use `File|Open ...` to get a standard file dialog box. The current folder will probably be the folder containing CodeWarrior. Move to the folder containing the desired project file and click `Open`.

- **The Project Window —Opening Files**

  When a project file is open, CodeWarrior displays the file's *project window*. This window lists all the files and libraries included in the project. To open an editor window for a source file, double-click the file name in the project window. (You may have to first click on the triangle (or '+' symbol) to the left of a group name to see the names of files in that group.)

5 Basic Program Editing

The CodeWarrior editor works much like any other Macintosh or Windows text editor. Source files are displayed in *editor windows*. The mouse is used to select text by clicking and dragging. Use the `Cut`, `Copy`, and `Paste` commands under `Edit` to move text around. `Undo` can reverse the effect of many commands if you make a mistake.

The most noticeable difference from most editors is the way CodeWarrior displays programs on a color monitor. Program code, keywords, and comments are displayed in different colors. This effect is called *syntax coloring*. Another visual effect that the editor provides is *balancing*. Whenever you type `)`, }, or `]`, the editor will momentarily highlight the corresponding symbol `(`, {, or `[`. If the highlighted left bracket is not the one you expected, or if the editor beeps because it can't find a matching bracket, check for extra or missing brackets.

There are editor commands to help indent code. The `Tab` key increases the indentation level, while `Backspace` decreases it. When you hit `Return`, the editor starts the new line with the same indentation as the previous one. To change the indentation of whole sections of code, select the lines and use `Edit|Shift Left` or `Edit|Shift Right`.

To save source files, use `File|Save`. CodeWarrior saves project files automatically; you don't need to explicitly save them.

> **IMPORTANT:** *Be sure to save your work frequently, particularly if you are creating a large program or have made many changes. Don't wait until you are done for the day. A power failure or machine crash could wipe out hours of work if you haven't saved files recently. You should also make a backup on a* ***separate*** *disk in case your master disk becomes unusable.*

6 Running Programs

To compile and run your program, select `Project|Run` (`Command-r` (Mac) or `F5` (Windows)).

If CodeWarrior detects errors in your program, it will display appropriate messages. To get to the trouble spot in the code, double-click an error message.

If no errors are found, CodeWarrior will start the Java Virtual Machine and instruct it to run your program. A console window will appear (this is where text output is written). If your program uses other windows, say for drawing, these will also appear. If they are hidden behind other windows, you can bring them forward by clicking on any visible part of the window or (Windows only) clicking on the window name at the bottom of the screen.

When the last statement in a Java program has been executed, the Java environment continues running to let you examine the program's output and any other program windows remaining on the screen. When you are done, you should quit the Java runtime environment by selecting `Apple|Quit` (Macintosh; the "Apple" menu is the leftmost, apple-shaped menu) or clicking the console window's `Close` box (Windows).

> *Remember to quit the Java environment before you select* ***`Project|Run`*** *in CodeWarrior again, especially if you make changes to your Java program. Otherwise, you may see problematic behavior such as (a) multiple copies of the program running, thus confusing you as to which copy has the changes, (b) running an old version that does not include your changes, or (c) an obscure error mumbling about a "Link" or "Post Linker Error". Quit the Java environment (not CodeWarrior) and select* ***`Run`*** *again to run the new version.*

7 Errors, Bugs, and Crashes

Debugging. As you develop programs, you are almost certain to encounter errors, including incorrect results, nonterminating execution, or (rarely) machine crashes. When this happens, the Metrowerks debugger can be used to help locate problems. The debugger will be described in a later handout.

Macintosh Freezes. Occasionally, an error in a program will cause the Macintosh to hang —everything stops and the mouse and keyboard don't have any effect. In this case, first try to force the program to quit by typing `Option-Command-Esc`. If that works, save any open files and restart the machine to reinitialize the Mac.

If the forced quit doesn't work, the only solution is to restart the machine. On older Macs, the restart button is the leftmost of the two located on the left side of the front panel. If your Mac doesn't have a restart button, try holding down the `Control` and `Command` keys and then pressing the `power-on` button at the top of the keyboard. When you release these keys, the machine should restart.

> **Warning:** *Don't restart unless there is no other option. You will have to restart all applications that were running, and you will lose changes to documents that were modified but not saved.*

Windows Freezes. While Windows doesn't lock up very often, we have seen situations where the Java environment starts behaving strangely, usually after some other problem has occurred. The best solution is to select `Start|Shutdown...` then select `Restart the Computer` to reinitialize Windows.

8 Printing

Once you have a correctly running program you will want to print a copy of the program and its output, perhaps because you need to hand it in for credit.

- **Printing in CIT Labs**

  Some printers in the labs are shared among three or four computers. Be sure nobody else is using the printer; then, turn the rotary switch connected to the printer to select your machine. Check that the printer is on and ready.

  You can also use NetPrint to print on a laser printer (there is a charge for this). See the CIT documentation for instructions.

- **Printing Programs and Other Files**

  Java source programs and other files can be printed from CodeWarrior the same way that text files are printed in a word processor. Select the CodeWarrior window containing the file, then select `File|Print...`. This brings up the standard print dialog box. If you're using an inkjet or dot-matrix printer, use either `Faster` or `Best` quality —`Draft` quality can be almost illegible. Click `Print` to print the file.

  CodeWarrior can also print part of a file. Select the lines you want to print, then check the `Print Selection Only` box in the `Print...` dialog.

- **Printing Program Output**

  Java and its class libraries were designed for use in web applets, which run in a browser, or embedded computers that control gadgets like appliances or smart credit cards. In these environments, there is little need for printed output, and Java provides little direct support for it.

  It is, however, possible to get printouts of the console window or a picture of the entire screen indirectly by creating files containing the desired output.

- **Printing Console (Text) Output**

  The basic idea is to select the text in the console window and copy it to a text editor, then use the text editor's print command to print it. The output will look best if a monospaced font like Courier or Monaco is used in the text editor.

  Macintosh. Select the console window, click at the bottom of the output, and drag the mouse to the top of the window. If text has scrolled off the top of the window, drag the mouse to the window's title bar and hold the mouse button down to scroll the window and select the text. Then select `Edit|Copy`. Paste the text into an editor window (a new CodeWarrior editor window will do fine) and print it.

  Windows. Click the right mouse button on the console window's title bar. Select `Edit|Mark` from the pop-up menu that appears. Click at the beginning of the text you want to copy, press and hold `shift`, then click at the end of the text. Click the right mouse button in the title bar again and select `Edit|Copy` in the popup menu. Click on a text editor window (a new CodeWarrior editor window, for example) and use `Edit|Paste` to copy the text. Print the window.

- **Printing Screen Snapshots**

  Both Macintosh and Windows have the ability to create a file containing a picture of the entire screen. Among other things, this provides a convenient way to print graphics output. First, arrange the windows on the screen so the ones you want to include in the printout are visible, then do the following.

  Macintosh. Press `Command-Shift-3` to take a snapshot of the screen. The output will be placed in a file named `Picture 1` in the main window of the hard disk. If a file named `Picture 1` already exists, the file will be named `Picture 2`, `Picture 3`, etc.

  To view the snapshot, double-click the picture file to launch `SimpleText` to display the file. To print the snapshot, select `File|Print...` in `SimpleText`.

  Windows. Press `Alt-PrintScrn` to take a snapshot of the currently active window, or `PrintScrn` to take a snapshot of the entire screen. Open `Paint` from menu `Start|Programs|Accessories`. Select `Edit|Paste` to copy the screen image to `Paint`, then print it.

9 Creating New Projects

For some programming assignments you will receive pre-built project and source code files. But if no files are supplied, or if you want to run additional programs for practice, you will need to know how to create your own projects.

To create a new project from scratch, start CodeWarrior and select `File|New...`. In the window that appears, select `Java Stationery` as the kind of project to create, and give the project file a name and choose where to put it. Type a name ending in `.mcp` for the project. To change where to save the new project, click `Set`; note that if you have not already selected a project name, you will be asked to do so in the resulting dialog box. Finally —after selecting `Java Stationery` as the kind of project, supplying a project name, and choosing where to save the new project— click `OK`.

In the next window that appears, select the appropriate *project stationery*. The different project stationery choices specify appropriate options, libraries, and sample code files needed to create different kinds of programs. CodeWarrior comes with several project stationery choices. In the CIT labs, we have added a

few extra choices to create simple Java programs for Computer Science classes. (See the end of this handout for information about adding them to your copy of CodeWarrior.)

If you are creating a simple Java program that does text input and output, select `CUCSJavaApplication`. If you want to generate graphical output, select `CUCSJavaGraphicsApplication`. Click `OK`. If a window pops up warning you that various project elements have "... been updated to the latest version", ignore it (you may close the window).

This creates a new folder in the location you selected earlier. The folder contains the new project file, copies of other files included with the stationery, and a folder containing data used by CodeWarrior for its internal bookkeeping. Once the files and folders are created, CodeWarrior will open the new project file.

> **IMPORTANT:** *When you save a new file or folder,* ***always*** *check that you are saving it where you want it. The default folder initially displayed in the dialog box is usually somewhere in the folders containing CodeWarrior itself, which is almost certainly* ***not*** *where you want to store your project.*

10 Working With Projects, e.g. Setting the Target

Once you've created or opened a project, you can modify the source file(s) supplied with the project stationery to create your program. You may also want to add additional source files to the program or remove some of the ones that are already there.

> *WARNING: Although you can have multiple projects open at the same time, this is not encouraged, because it is extremely easy to get confused among the projects.*

- **Creating New Java Source Files**

  Java programs are built from a collection of classes. Each class should normally be stored in a file whose name is the class name followed by `.java` (e.g. class `DrawPicture` should be stored in file `DrawPicture.java`.

  To create a new file, select `File|New Text File`, enter some Java code in the new window, then save the file with the appropriate name. Double-check to be sure you save it in the correct folder, which should usually be the one that contains the project file.

- **Adding Java Files to Projects**

  Before a Java file can be used as part of a program, it must be added to the project window —just creating a `.java` file isn't enough. You can add to the project files that you create or files containing Java code that you get from us or other sources.

  Adding open windows. If the Java file is open in a CodeWarrior editor window, bring the window to the front, then select `Project|Add Window`.

  Adding other files. To add other Java files to the project, either go to the desktop and drag the files into the project window, or else select `Project|Add Files...` . This brings up a dialog window, which you use to pick the files. Find the Java file(s) by moving to the correct folder(s) if necessary. Select the file name(s) in the top window and click `Add`. The selected file names should appear in the list at the bottom of the window.

  You can select as many files from as many folders as you like. When you're finished, click `Done` to add them to the project.

- **Rearranging Project Files**

  The files in the project window can be arranged in any order you want. To move a file, drag it to the desired place.

- **Selecting Method main (Selecting the Target Class)**

Every Java application program begins execution in method `main` in one of its classes. Since every Java class can have a `main` method (often a good place to put code to test the class), one has to specify which `main` should be used to start the program. When you create a new Java project from stationery, this is done automatically. For example, the `CUCSJavaApplication` and `CUCSJavaGraphicsApplication` stationery begin execution in method `main` of class `Application` or `GraphicsApplication` respectively.

If you want to begin execution in another class (perhaps you've added one to the project or renamed the class used, e.g. `CUCSApplication` to `Prog7`), you **MUST!!** change the project settings to identify the class containing the appropriate `main` method. Click on the leftmost rectangular icon in the middle of the project window or select `Edit|Settings...`. (Note: The menu item will have some kind of project name in front of "Settings", e.g. `Java Application Settings`.) This opens the settings panel. Select `Java Target` in the list of options at the left of this window. Enter the name of the **class** —not the name of the file— containing the desired `main` method in field `Main Class:`. Click `Save`, then close the window.

> **IMPORTANT:** *If the `Main Class` setting is wrong, CodeWarrior may exhibit very bizarre behavior when you try to run the program. If you're lucky, you'll get an error message complaining that some class you've never heard of can't be found.*
>
> *However, the Java Virtual Machine may begin executing whatever code it can find, strange windows may appear on the screen, or you may get error messages from Windows itself complaining about system-level issues that seem totally unrelated to your Java program. If you see such bizarre behavior, the `Main Class:` setting is one of the first places to look for trouble.*

- **Removing Files From Projects**

  You can remove files from the project window if you no longer need them, e.g. to remove files included with the stationery (like `Application.java` or `GraphicsApplication.java`) when you want to substitute another class containing a different `main` method.

  To remove a file, select the file name in the project window, then use one of the following techniques: (PC) hit the Delete key; (Mac) hit the Delete key or press Command-Delete; (PC only) right-click on the file name and select `Delete` from the resulting pop-up menu; (PC or Mac) select `Project|Remove` (this menu item doesn't always exist!).

- **Changing Project Settings**

  You may want to change some of the other CodeWarrior settings for your project. You can change cosmetic things like the editor's font, text size, and keyword colors, as well as more substantive things like compiler options. Select `Preferences...` or `Settings...` from menu `Edit` to find out what's available. You can change things like the editor colors and fonts freely, but be careful about altering options you don't understand. Inappropriate changes can make it impossible to run your program.

  > *One mysterious option that you might see on a PC is for all of CodeWarrior's windows to be inside a single "master" window. If this happens and you wish to turn it off, select `Edit|Preferences`. In the resulting IDE Preferences window, select the IDE Extras panel and uncheck "Use Multiple Document Interface".*

- **Project File Compatibility**

  Program (`.java`) files should be compatible across versions of CodeWarrior and be compatible between the Mac and PC, and project (`.mcp`) files should be compatible between the Mac and PC. However, CodeWarrior Pro 5 uses a new internal format for project files. While old project files can be updated —this might be somewhat tricky on the Mac— to work with this release, new project files, including ones you get from us, will not work with previous versions of CodeWarrior. We suggest you upgrade to the current version, although you can create a new project file with an older version and add to it the source files you get from us.

- **Damaged Project Files**

  Occasionally, usually after a particularly bad crash, a project file becomes damaged and CodeWarrior behaves strangely when using it. If the project file is badly damaged, CodeWarrior might crash when `Run` is selected —even before the program gets a chance to crash on its own— or changes in the source program seem to have no effect on what happens when the program is executed. (But be sure this is not because you forgot to quit the Java runtime environment after your program finished.)

  The first step when you suspect the project file is damaged is to force CodeWarrior to recreate some of the information in it. Select `Project`|`Remove Object Code`. This command causes CodeWarrior to discard the machine-language version of the program and some other information in the project file. This information will be recreated the next time you try to run the program.

  If this doesn't solve the problem, the only solution may be to discard the damaged project file and use a new one. Get a new copy of the project file, if it was supplied as part of an assignment, or create a new project from stationery; add the program's other `.java` files to the project; check the `Main Class:` option; and delete from the project any stationery files you don't need.

11 Editing Hints

The CodeWarrior editor has many features to help edit programs. Here are some useful ones.

- **Opening Related Files**

  There are several ways besides `File`|`Open...` to open files associated with a project.

  To open a file whose name appears as text in an editor window, select the name of the file and select `File`|`Open Selection`.

  If you are editing a Java class, you can display a menu of files containing other classes used by that class by pressing the button containing an `h` in the upper left corner of the editor window. Another way to do this is to press the triangle to the right of the file name in the project window. In either case, you'll see a list of files containing related classes; select a name from the list to open the file. (This doesn't work until the program has been compiled at least once. CodeWarrior builds the list of related files when it compiles a file.)

- **Searching**

  The editor window provides a quick way to go to the definition of any function (method) in the current file. Press the {} button in the upper left corner to see a list of functions in the file. Choose a function name to jump to the definition of that function.

  To find text strings, use the *find window*. Select `Search`|`Find...` . The window that appears contains text boxes for the *find* and *replace* strings. As in any text editor, when you enter a find string and click `Find`, CodeWarrior selects the next occurrence of that string. Click `Replace` to replace it with the string in the *replace* field.

  These commands provide the basic operations for searching in source files. Experiment with the other commands under `Search` as well. Some of the commands can be modified by pressing the `Shift` key. For example, `Shift` changes `Find Next` to `Find Previous`, which searches backward instead of forward through the file.

12 Getting More Information or Help

Web pages from Sun documenting the full Java class library can be reached from the CS100 web pages. This is a convenient way to look up information about standard Java classes.

Electronic versions of CodeWarrior manuals and other documentation are included on the CodeWarrior CD-ROMs. Be aware that the CodeWarrior documentation is written **by** experienced programmers **for** experienced programmers, so it may be quite cryptic or obscure in some places.

For CodeWarrior updates and information, check Metrowerks' Web site:

`http://www.metrowerks.com`

If you have questions about CodeWarrior, first contact staff members for your course —consultants, teaching assistants, or instructors.

If you find errors in this missive, please send email to `cs100@cs.cornell.edu` describing the problem.

Metrowerks provides email technical support for CodeWarrior. If you think you've found a CodeWarrior bug, you can contact them directly, but it would be best to check with a member of the course staff first to verify that there really is a bug and not some other problem.